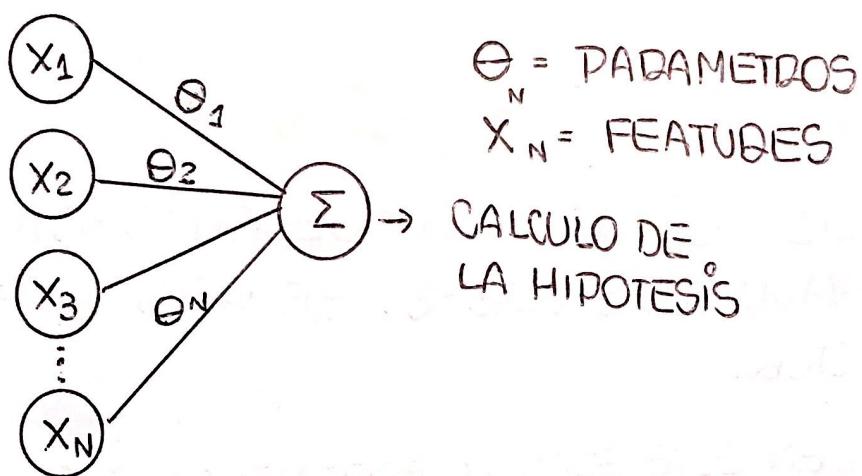


# REDES NEURONALES

## REGRESIÓN LINEAL

- ✓ RESUELVE PROBLEMAS DE REGRESIÓN. EL MODELO INTENTA AJUSTAR UNA RECTA AL CONJUNTO DE DATOS.
- ✓ PUEDE SER LINEAL EN 'D' DIMENSIONES.

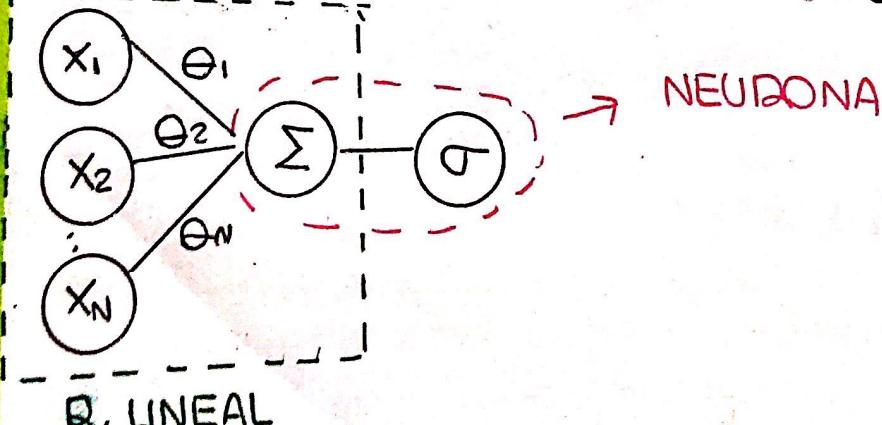


## REGRESIÓN LOGÍSTICA

- ✓ AGREGA LA REGRESIÓN LINEAL Y LA CONVIERTE EN UNA CLASIFICACIÓN BINARIA. ES DECIR, RESUELVE PROBLEMAS DE CLASIFICACIÓN.

PARA ESTO APLICAMOS UNA TRANSFORMACIÓN / FUNCIÓN QUE NOS PONGA LA REGRESIÓN LINEAL EN EL INTERVALO  $[0, 1]$ .

- ✓ USAMOS LA 'FUNCIÓN LOGÍSTICA' → **SIGMOIDE**

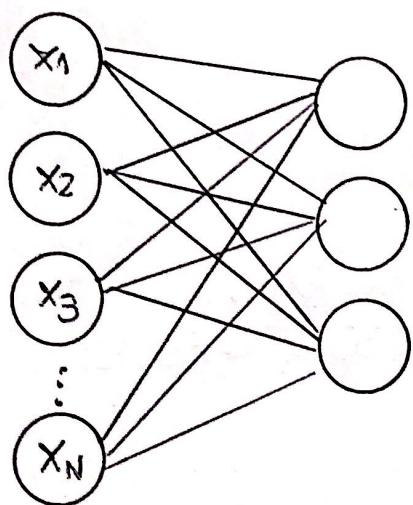


# PERCEPTRÓN

- ✓ ES LINEAL
- ✓ PUEDE UTILIZARSE DE MANERA ONLINE.
- ✓ TIENE CONVERGENCIA ASEGURADA
- ✓ ES IGUAL A LA REGRESIÓN LOGÍSTICA PERO UTILIZA LA FUNCIÓN DE ACTIVACIÓN **SIGNO**

# REGRESIÓN SOFTMAX

- ✓ ES UNA GENERALIZACIÓN DE LA REGRESIÓN LOGÍSTICA PARA EL CASO DE MÚLTIPLES CLASES. SE DESCRIBE EL CASO BINARIO COMO VECTOR.



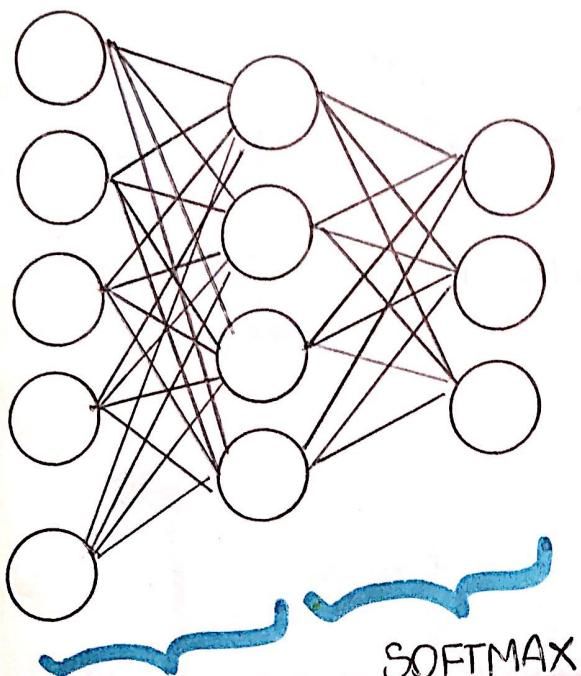
EN ESTE EJEMPLO SUPONEMOS 3 CLASES, TENDRÉ TANTAS NEUDONAS COMO CLASES. SON COMO UN VECTOR. LA PRIMERA NEUDONA TIENE LA IP DE QUE PERTENEZCA A LA C1 Y ASÍ ...

FUNCIÓN DE ACTIVACIÓN **SOFTMAX**. DEVUELVE PROBABILIDADES Y TE QUEDAS CON LA MÁS ALTA.

✓  $\sum$  DE LAS NEUDONAS = 1.

# PERCEPTRÓN MULTICAPA

- ✓ INICIALMENTE EL PROBLEMA NO ES LINEAL
- ✓ LE APLICAMOS UNA FUNCIÓN AL SET PARA MOVER LOS DATOS YA SEA EN OTRAS DIMENSIONES PARA PODER SEPARAR LINEALMENTE.



MOVEMOS  
LOS DATOS  
APRENDE AQUÍ ES  
EL MAPEO PARA QUE  
LAS CLASES SEAN  
SEPARABLES.

- ✓ PUEDO TENER MUCHAS CAPAS O UNA SOLA, ESTE ES UN H-P

QUE HAY QUE BUSCAR EL VALOR ÓPTIMO.

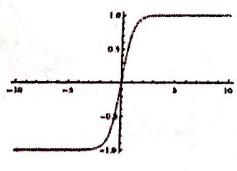
## TEOREMA DE APPROX UNI

ESTE TEOREMA NOS GARANTIZA QUE CON UNA SOLA CAPA CON SUFICIENTES NEURONAS PODEMOS SEPARAR CUALQUIER SET DE DATOS.

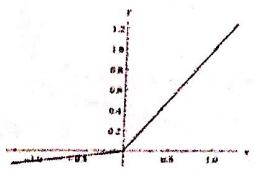


HAY QUE TENER CUIDADO, PORQUE CUANTO MÁS CAPAS, HAY MÁS PARAMETROS PARA APRENDER, ENTONCES NECESITAMOS TENER MUCHOS DATOS PARA EVITAR EL OVERFITTING

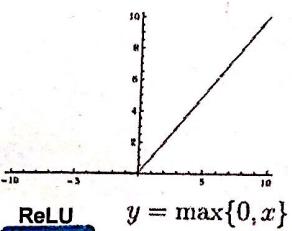
En este contexto  $\sigma$  se llama función de activación. Hasta ahora usamos la función sigmoide, pero hay otras:



tanh



LeakyReLU



ReLU

$$y = \max\{0, x\}$$

MUY POPULAR

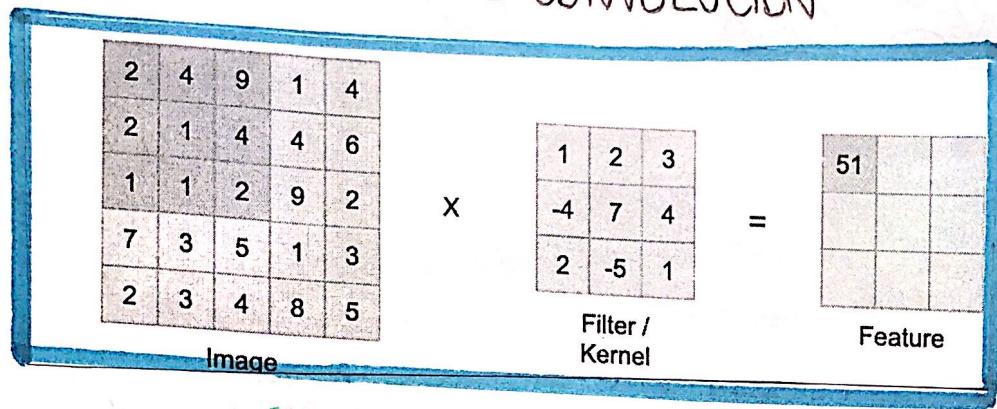
- ✓ SE ENTRENAN CON BACK PROPAGATION

# REDES CONVOLUCIONALES

- ✓ SE UTILIZAN PARA CLASIFICACIÓN DE IMAGENES
- ✓ PARA PODER PASARLA LA IMAGEN A LA RED, SE UTILIZAN DISTINTOS **FILTROS**

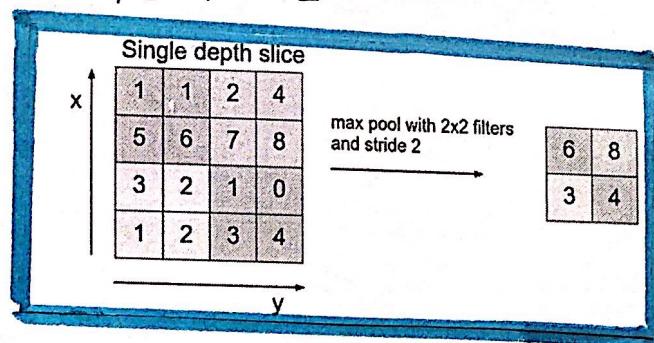
## FILTRO KERNEL

- ✓ ESTE FILTRO REDUCE EL TAMAÑO DE LA IMAGEN, ESTO IMPLICA QUEDARSE CON LAS CARACTERÍSTICAS MÁS IMPORTANTES DE LA IMAGEN.
- ✓ SE LO DENOMINA CAPA DE CONVOLUCIÓN



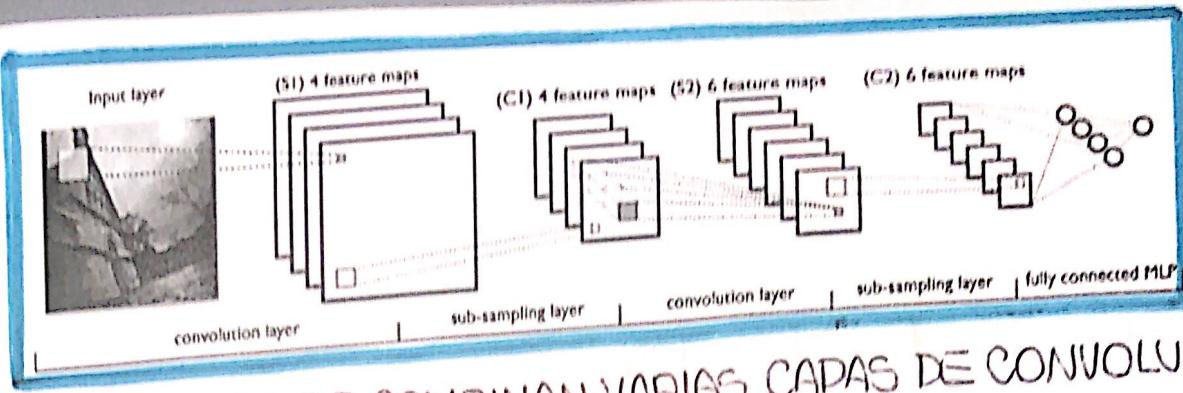
## MAX-POOLING

- ✓ PARA DISTINTAS SECCIONES DE IMAGEN SE OBTIENE LA FEATURE MÁS IMPORTANTE.



## HIPER-PARAMETROS

- ✓ CANTIDAD DE FILTROS A USAR
- ✓ TAMAÑO DE FILTRO
- ✓ STRIDE (PASO)
- ✓ PADDING

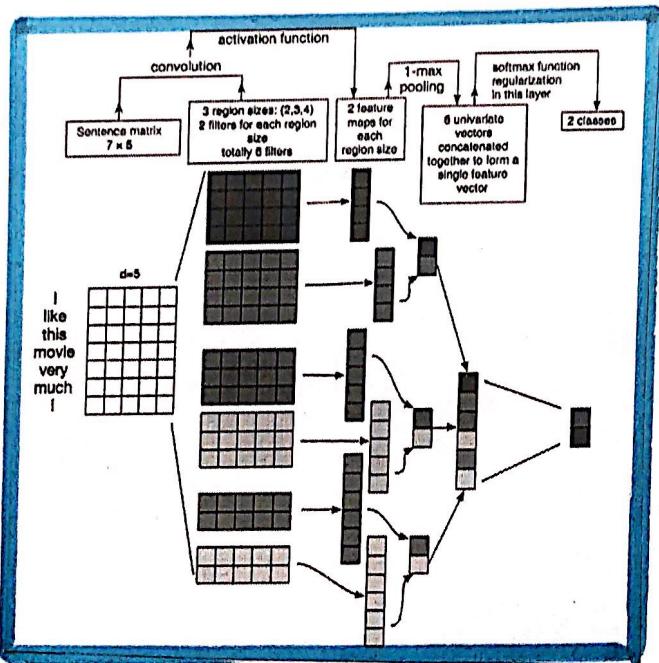


ENTONCES SE COMBINAN VARIAS CAPAS DE CONVOLUCIÓN Y  
DE MAX-POOLING PARA IR REDUCIENDO LA IMAGEN Y LUEGO  
SE ALIMENTA A UNA CAPA DENSA DE RED NEURONAL. NO  
OLVIDAB QUE ANTES DE ESO LA IMG SE APLASTA A 1.D.

# CONV1D

✓ ES POSIBLE APLICAR EL MISMO CONCEPTO QUE USAMOS PARA IMAGENES EN TEXTO.

✓ REPRESENTAMOS A CADA PALABRA DEL TEXTO COMO UN VECTOR DE  $N$  DIMENSIONES MEDIANTE **EMBEDDING**



ESTO SE HACE CON CADA  
REGISTRO.

# REDES PROFUNDAS

- ✓ BUSCAN ABSTRAERSE DEL PROBLEMA DE UTILIZAR MAS DE DOS CAPAS
- ✓ CADA CAPA SE ABSTRAE MAS DE LOS DATOS INICIALES, OBTENIENDO MEJOR REPRESENTACIÓN DE LOS DATOS.
- ✓ SE PRESENTAN ALGUNAS DIFICULTADES AL ENTRENAR SI LAS CAPAS NO APRENDEN AL MISMO TIEMPO, (RITMO).
- ✓ PARA EVITAR EL OVERFITTING SE PUEDE USAR:
  - DROPOUT
  - BATCH
  - REGULARIZACIÓN

